DOCUMENT RESUME

ED 143 371 IR 005 161

AUTHOR Fletcher, J.D.; Zdybel, Frank
TITLE Intelligent Instructional Systems in Military Training.
PUB DATE Apr 77
NOTE 17p.; Paper presented at the annual meeting of the American Educational Research Association (New York, New York, April 1977)

EDRS PRICE MF-$0.83 HC-$1.67 Plus Postage.
DESCRIPTORS *Artificial Intelligence; *Computer Assisted Instruction; Computer Oriented Programs; *Instructional Design; Instructional Systems; *Military Training

ABSTRACT

Intelligent instructional systems can be distinguished from more conventional approaches by the automation of instructional interaction and choice of strategy. This approach promises to reduce the costs of instructional materials preparation and to increase the adaptability and individualization of the instruction delivered. Tutorial simulation and tutorial dialogue capabilities require a computer to: (1) generate problem statements and solutions; (2) determine efficient sequences; and (3) simulate a variety of situations encountered on the job. These enable students to: (1) test their own hypotheses concerning the subject matter; (2) probe for information at different levels of difficulty and abstraction; (3) acquire wide experience in minimum time; (4) obtain instructional material generated for their unique abilities and needs; (5) receive instructional aids for partially completed solutions; and (6) receive reviews and critiques of completed problem solutions. Description of the Welfare Effectiveness Simulation (WES) in military training, directions for development of intelligent instructional systems, and references are included. (Author/KP)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

ED143371

Intelligent Instructional Systems in Military Training

J. D. Fletcher and Frank Zdybel

Navy Personnel Research and Development Center
San Diego, California 92152

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

PERMISSION TO REPRODUCE THIS COPYRIGHTED MATERIAL HAS BEEN GRANTED BY

John D. Fletcher

TO ERIC AND ORGANIZATIONS OPERATING UNDER AGREEMENTS WITH THE NATIONAL INSTITUTE OF EDUCATION. FURTHER REPRODUCTION OUTSIDE THE ERIC SYSTEM REQUIRES PERMISSION OF THE COPYRIGHT OWNER

IR005 161

Presented at American Educational Research Association Annual Meeting, April 1977, New York, New York.

SESSION 27.05

Intelligent Instructional Systems in Military Training [1]

J. D. Fletcher and Frank Zdybel
Navy Personnel Research and Development Center
San Diego, California 92152

Under its earlier institutional title of Naval Personnel and Training Research Laboratory, the Navy Personnel Research and Development Center compiled an extensive history of research in computer-based instruction (CBI), and, as one might expect, this work is continuing with investigations of state-of-the-art CBI techniques including those associated with PLATO and TICCIT technologies, both of which systems are in use at the Center. Overviews of this work have been presented by Ford, Slough, and Hurlock (1972), Fletcher (1975), and Hurlock and Slough (1976). A natural concomitant of this work is an interest in investigating promising CBI techniques such as optimization of instruction, learner control, computerized, adaptive, criterion-referenced testing, and intelligent instructional systems. In discussing intelligent instructional systems it is usually necessary to point out hastily that no distinction is intended between "intelligently" designed instructional systems and "unintelligently" designed instructional systems. Intelligent instructional systems may be as unintelligently designed as any others. Rather the development of these systems is viewed as a specific effort to apply artificial intelligence techniques to CBI in the sense of information structure oriented (ISO) approaches discussed and advocated by Carbonell (e.g. 1970). This paper discusses the capabilities of these systems, their uses for military instruction, and directions for their further development.

Capabilities

Doubtless, we are all familiar with examples of the speed and accuracy with which computers process vast amounts of information. Micro-seconds no longer seem appropriate for describing the speed of many machines; we are down to nano-seconds and the need for pico-seconds appears imminent. The accuracy of computers in accessing billions of bits of information without losing, or "forgetting," one remains impressive, and the amount of information a computer can process seems endless. A PDP-10 can examine 20 million bits in one second; it would take one of us about four months to do the same, assuming we could process two bits every second of every day, seven days a week. Still, we like to emphasize that computers are dumb--they only do what we tell them. In a sense, however, and independent of speed, accuracy, and vastness of storage, computers are capable of powerful forms of processing that we often neglect.

Consider the following sentence:

The man the dog the girl owned bit died.

This is a difficult sentence for people to parse; they are easily lost in its syntactic nestings. Yet a computer could very quickly discern, after diving into its recursive routines for processing nested constructions

that there was a dog that was owned by a girl, that the dog bit a man, and that the man subsequently died.

Here is another example:

The man kicked the ball kicked the ball.

This is a perfectly grammatical sentence as any self-respecting machine would discover after reversing an English transformational rule for deleting function words and discerning that a man to whom a ball was kicked, kicked the ball back. In both these examples, a computer is less likely than we are to be confused or distracted, and its ability to process these two examples illustrates real intellectual ability. "Artificial intelligence" is after all a poor name for the business of making computers intelligent. Intelligence, or intellectual ability, is really what the field is all about. That theories of intelligence are tested by algorithmization and putting them on computers is merely an issue of methodology, albeit a central one; there is nothing artificial about the capabilities targeted by this work.

These two examples also illustrate the distinction between what John Brown (e.g., Brown, Burton, and Bell, 1974) calls "intelligent" CBI and what might be called "knowledgeable" CBI. We all seem susceptible to the confusion of knowledge and intelligence. We expect someone who is intelligent to know that relativity probably has more to do with physics than with family ties, that battery may involve either an electrical or a legal charge, and that scales are relevant to both fish and music. However, there are many intelligent people who have not heard of the theory of relativity, the legal aspects of fisticuffs, or B-flat minor, just as there are many educated people whose inferential capabilities are less than awesome. A computer program that could process the two examples above would qualify as knowledgeable, and, at least in the second example, demonstrate a clarity of thought that is superior to most humans' and is independent of the speed, accuracy, and vastness of storage characteristic of computers.

Consider the following example, stolen from Donald Norman (1973):

What was Charles Dickens' telephone number?

A knowledgeable program would search the attributes it had associated with Charles Dickens looking for a telephone number, and/or it would search its list of telephone numbers looking for one associated with Charles Dickens. Presumably no telephone number will be found for Charles Dickens, and this fact will simply be reported. As most human information processors must be aware, there is a more intelligent answer to this query that completely avoids searching lists of telephone numbers and/or attributes. To reply that telephones were not used until after Dickens' lifetime does require some knowledge, but this information could easily be stored by a computer. The real problem is to devise a computer program that knows when this information is relevant.

Another example taken from Abelson and Reich (1969) is the so-called drugstore problem.

You might say:

I went to three drugstores.

A syntactically appropriate response would be:

How did you go to three drugstores?

A semantically appropriate response would be:

What did you buy in the drugstores?

However, the most natural response would be:

What were you trying to buy that the first two drug-stores didn't have?

This last response requires an understanding of what Abelson and Reich called the "pragmatics" of language--the penetration beyond literal decoding to plausible implications. The response also illustrates the distinction between a knowledgeable and an intelligent program. The implications of "pragmatics" are broader than Abelson and Reich foresaw; they transcend the problems of natural language processing and encompass all propositional and conceptual systems.

To sum it up, we have the kind of processing capability required by the first two examples and we should use it. We need the "intelligent" processing capability required by the last two examples, and it looks as if we can get it. Advances have been made in the capabilities of computers to process pragmatics. Notably, most of these advances have been made in the context of intelligent instructional systems.

Consider the following student/computer protocol taken from John Brown's SOPHIE system (Brown, Burton, and Bell, 1974). In this example, the student is completing a series of tests on a computer simulated piece of electronic equipment (a regulated power supply) with a fault generated by the computer. Student responses are underlined.

<u>WHAT IS THE BSE EMITTER VOLTAGE OF THE VOLTAGE LIMITING TRANSITOR?</u>

BASE
TRANSISTOR
THE BASE EMITTER VOLTAGE OF Q5 IS - .037 MILLIVOLTS.

The spelling corrector has detected and corrected two misspellings. The system must also know that Q5 is the voltage limiting transistor.

<u>WHAT ABOUT FOR THE CURRENT LIMITING TRANSISTOR</u>

THE BASE EMITTER VOLTAGE OF Q6 IS .349 VOLTS.

The meaning of this question is completely dependent on the context established by the prior sentence. The system uses this context to determine that he is interested in not only the voltage measurements but the voltage across a particular junction.

WHAT SHOULD IT BE?

IN A WORKING CIRCUIT THE BASE EMITTER VOLTAGE OF Q6 IS .593 VOLTS.

Another way of requesting the proper measurement to be repeated is a working instrument, i.e., the meaning of "should" implies a working instrument should be used.

REPLACE Q6

I AM GOING TO ASK YOU SOME QUESTIONS ABOUT HOW Q6 IS FAULTED. ARE ANY JUNCTIONS SHORTED?

Before the system replaces a component it queries him as to what he thinks is wrong. If he gets at least something right, the system tells him what was actually wrong and goes ahead and replaces it.

I DON'T KNOW

ARE ANY TERMINALS OPEN?

YES

WHICH TERMINALS ARE OPEN?

THE BASE

Q6 IS NOT FAULTED IN THAT MANNER. Q6 NOT REPLACED.

IS IT POSSIBLE THAT Q6 HAS AN OPEN BASE?

The student now has an idea what might be wrong. However, before the system's evaluation procedures can be applied to his idea or hypothesis, the hypothesis must be completely specified. To help the student out, the system suggests a plausible complete specification of it.

IS THE COMPLETE FAULT MODE EB/EC/BC = OP/OK/OP WHAT YOU INTENDED?

SURE

WHEN THE SETTINGS WERE CC=1.0. CR=HIGH, LOAD=20, VC=1.0,
VR=HIGH THAT FAULT CONFLICTS WITH THE FOLLOWING MEASUREMENT.
YOU OBSERVED THAT THE OUTPUT VOLTAGE WAS 11.7 VOLTS. IF Q6 HAD
FAULT MODE EB/EC/BC = OP/OK/OP IT WOULD BE 23.9 VOLTS. IN A
WORKING CIRCUIT IT IS 19.9 VOLTS.

In determining the logical consistency of his hypothesis with respect to the measurements he has already made, a glaring counter-example is discovered.

ALSO:
THAT FAULT CONFLICTS WITH THE FOLLOWING MEASUREMENT.
YOU OBSERVED THAT THE BASE EMITTER VOLTAGE OF Q6 WAS .349 VOLTS.
IF Q6 HAD FAULT MODE EB/EC/BC = OP/OK/OP IT WOULD BE .712 VOLTS.
IN A WORKING CIRCUIT IT IS .593 VOLTS.

The system then points out a more subtle counter-example.

THAT FAULT DOES NOT EXPLAIN ANY OF THE SYMPTOMS YOU OBSERVED BUT IT IS CONSISTENT WITH THE VALUES FOR THE BASE EMITTER VOLTAGE OF Q2 AND THE VOLTAGE AT N1.

Finally the system tells him what is right or at least consistent with his hypothesis.

Clearly, SOPHIE has progressed beyond knowledgeable syntactic and semantic processing of student inquiries and is behaving in an "intelligent" manner. The system tries to create a reactive environment in which the student learns by actively trying out his ideas in interaction with a simulation program. However, the program does more than simply simulate the information to be transmitted, it provides for tutorial feedback and, in effect, for a one-to-one relationship with an "expert" problem solver who helps the student create, experiment with, and debug his own ideas.

Consider another student/computer protocol obtained from the Stanford University CAI course in elementary mathematical logic. The protocol was documented by Goldberg (1973). Problems in this course are not generated by computer, but their solutions are. The course thereby achieves substantial flexibility in providing hints for partial solutions and critiques for full solutions. The following abbreviations are used for axioms:

CA	Commute Addition	A+B=B+A
Z	Zero Axiom	A+0=A
AI	Additive Inverse	A+(-A)=0
N	Negative Number Axiom	A+(-B)=A-B

As in Brown's example, the student's responses are underlined; $ and : are delimeters.

:AI$	A + (-A) = 0
A::A$	(1) (A + -A) = 0
:N$	A + (-B) = A-B
A::A$	
B::A$	(2) (A + -A) = (A - A)
:HELP$	TRY TO SHOW THAT 0 + -0 = -0
:Z$	A + 0 = A
A::-0$	(4) (-0 + 0) = -0
:HELP$	DID YOU CARRY OUT THE FIRST SUGGESTION?
:YES$	NO YOU DIDN'T. LOOK CAREFULLY AT LINE 4.
:CA1$	(5) (0 + -0) = -0
:HELP$	DO YOU SEE WHAT TO DO IF YOU SUBSTITUTE 0 FOR A IN THE AI AXIOM?
:YES$	OK, DO IT.
:AI$	A + (-A) = 0
A::0$	(6) (0 + -0) = 0
:5.6RE1$	(7) 0 = -0

CORRECT...

The subject matter in this example is only of incidental interest. The significant aspect of the example is its generation of tutorial guidance

based on the program's interpretation of the subject matter. As in Brown's example, the program is behaving intelligently; it is making decisions and interpretations well beyond its store of factual knowledge.

Uses for Instruction

The power and pedagogical utility of systems such as Brown's and Goldberg's should be self evident. However, as some reviews have indicated, we may be doing well enough with the CBI techniques we have. One might argue that a more appropriate investment strategy for instructional research and development is to continue work on already available techniques rather than attempt to yoke together artificial intelligence techniques and CBI. As researchers we have only one response; we attempt these developments because they are there, or we think they are. As practitioners with operational committments and funding sources to satisfy, there are at least three arguments we can make for development of these techniques.

First, they will provide a means for reducing the costs of instructional materials preparation. A difficulty that has inhibited the progress of both programmed instruction and the most common types of computerized instruction is the expense incurred in developing instructional materials for these media. One way to reduce this expense is to automate the production of items, branching strategies, and hints. This automation is possible, but it requires adequate models of the subject matter being taught. These models will become practical and available if current artificial intelligence techniques for representing knowledge can be made sufficiently general to apply to a variety of subject areas. Once these models are available, instructional materials can be generated for presentation either on-line, by computer, or off-line.

Second, they provide systematic, explicit bases for adapting instruction to the capabilities and needs of individuals. Considerable efficiencies in instruction, including the reduction of training time, will be realized by taking account of what the learner knows, concentrating on instructional material that he has not mastered, and emphasizing instructional strategies that are most effective for his cognitive style. Instruction that is uniquely adapted to individual learners has been receiving increasing emphasis. Adaptive instruction of this sort is based on adequate models of the learner. In the current state-of-the-art, these models are inadequate without references to a specific subject matter. A model of subject matter represents a learner when those aspects of the model that the learner has mastered are explicated. Such explication should be an integral part of an intelligent system for CBI.

Third, they will explicitly support simulations for a variety of instructional experiences. Many desirable training situations are impracticable because of their expense in terms of the time or the complexity of equipment they require.

A variety of intelligent instructional systems have now passed through initial development. Suppes and Goldberg's mathematical logic system and John Seely Brown's electronic troubleshooting systems were discussed above

and are documented by Goldberg (1973), Goldberg and Suppes (1974), Brown, Burton, and Bell (1975) and by Brown, Rubinstein, and Burton (1976). Other systems that deserve mention are the BASIC Instructional Program for teaching computer programming in BASIC (Barr, Beard, and Atkinson, 1975), Kimball's (1973) program for teaching integration techniques in calculus, the Carbonell/Collins SCHOLAR system for teaching South American geography (Collins, Warnock, and Passafiume, 1974), Koffman and Blount's (1975) MALT system for teaching assembler language programming, and Ruth's (1974) system for examining programs written by students in a restricted ALGOL-like language. Presumably these systems will find applications in both military and civilian training. One system under development that is likely to be applied only to military training and is particularly relevant to this discussion is the Warfare Effectiveness Simulator (WES).

WES, as it is being developed by the Navy Personnel Research and Development Center, simulates primarily physical aspects of tactical encounters between two or more opposing naval forces. At the simplest level, the program is a functionary that accepts orders issued by its users to simulated combat units, maintains a representation of the current state of affairs, and generates reports that describe conditions and events within the model. In accepting orders from the user, the program adheres to the behavioral and physical limitations of the situation being simulated. Similarly, in generating reports to the user, it provides only information that is accessible by simulatable means. The program is therefore a gaming medium that represents the current state of affairs, provides appropriate opposition to the student-player, and performs some of the functions of a referee. WES is intended to provide real-time interactive experience for decision making and tradeoff analyses within a wide range of realistic and dynamic tactical environments.

The naval forces within WES are divided into five classes of assets: weapons, sensors, ships, aircraft, and shore bases. Parameter categories assigned to these assets are the following:

1. Weapons

    a. Name
    b. Type (i.e., ammunition, bomb, torpedo, surface-to-air missile, air-to-air missile, etc.)
    c. Maximum range
    d. Probability of success at maximum range (less than 50%)
    e. Range at which a 50% probability of success occurs
    f. Number of rounds
    g. Rate of fire (per minute)
    h. Mean Time Between Failure
    i. Blast power (TNT equivalent)
    j. Susceptibility of the weapon to jamming

2. Sensors

    a. Name
    b. Type (i.e. radar, sonar, stationary, underwater listening device, etc.)
    c. Maximum detection range

d. Probability of detection of a target at maximum range (less than 50%)
e. Range at which 50% probability of detection of a target occurs
f. Susceptibility to jamming
g. Mean Time Between Failure
h. TNT equivalent blast which will destroy or render the sensor inoperative

3. Ships

a. Class name (i.e., CV65, DLG1, etc.)
b. Type (i.e., CV, DD, SUB, etc.)
c. Maximum speed (knots)
d. Cruising speed (knots)
e. Range, i.e., maximum distance realizable at cruising speed
f. Gallons of fuel oil carried
g. Gallons of J.P. fuel carried
h. TNT equivalent blast that renders the ship inoperative
i. Names of sensors aboard the ship (up to 3 sensors selected from the list defined in the sensor section above)
j. Name and number of weapons carried (up to 9 types selected from the list defined in the weapon section above)
k. Name and number of aircraft carried (up to 9 types selected from the list defined in the aircraft section below)

4. Aircraft

a. Class name (A4, F8, P3, etc.)
b. Maximum altitude (feet)
c. Maximum speed (knots)
d. Cruise speed (knots)
e. Maximum range at cruise speed
f. Gallons of fuel carried
g. Names of sensors on board (up to 3)
h. Names and numbers of weapons carried (up to 3)
i. TNT equivalent blast that destroys or renders the aircraft inoperative

5. Shore Bases

a. Name
b. Type (i.e., NAVSTA, AIRSTA, combined NAV/AIRSTA, etc.)
c. Geographical location (latitude and longitude)
d. TNT equivalent blast that destroys the base
e. Names and numbers of weapons (up to 9 types selected from the weapon list)
f. Names of sensors (up to 3 types selected from the sensor list)
g. Names and numbers of aircraft (up to 9 selected from the A/C list)
h. Gallons of fuel oil available in supply
i. Gallons of jet propulsion fuel available

Orders and suborders presently available in WES are:

a. MANEUVER: Allows the player to change course and speed of a platform or group of platforms. It also allows him to change the altitude of an aircraft.

b. A/C LAUNCH: Allows launching of aircraft from appropriate ship or shore bases. This order includes an initial course, assigned altitude and return (BINGO) time, as well as flight identification.

c. A/C RECOVER: Allows recovery of aircraft by carriers.

d. CHANGE DESTINATION: Allows changing the destination base of an aircraft or a flight.

e. EW ON: Allows passive sensors to be turned on and detection reports provided to the player.

f. REPORT: Allows active sensors aboard specified platforms to be turned on and detection reports provided to the player.

g. ATTACK: Allows attacks to be conducted between platforms or platforms and bases as specified in the order.

h. PLOT: Provides a plot of specified platforms on a grid of longitude/latitude lines.

i. CANCEL ORDER: Cancels any specified order.

j. END: Ends the game.

WES is based on and made possible by a model of tactical activity. The idea underlying the structure of the model is the division of the modeling activity into co-processes which are executed within the context of particular platform records. This approach permits the total behavior of the model to be sectioned into smaller, more manageable portions whose interactions can be limited, and it permits the behaviors of entities in the model to be varied radically without proportionately radical increases in the operations of the model executive. One of the weaknesses in dividing processes according to orders is seen most particularly in the attack process which attempts to express as a single activity an involved sequence of events in which a target is selected, preconditions to the attack are checked, and a weapon subsequently deployed. In order to permit meaningful variation in the attack behavior of platforms, these constituent activities should be made separately available as independent responses. Current efforts are underway to re-organize the model to follow the customary divisions of ships. One process, analogous to a ship's Combat Information Center, would be responsible for coordinating orders to the other departments to produce realistic mission-like behavior. Another process would simulate the physical and kinetic behavior of the platform in response to rudder and throttle changes. A third process would simulate the electronic warfare activities of operations departments. Other processes will be developed in a like manner. This new organization is expected to be more appropriate for smaller scale tactical training such as that required for command and control of a single platform rather than an entire task force. This organization will also enhance the intelligence of platforms as discussed later.

WES currently spends the bulk of its time in characterizing interactions between platforms, and the amount of time spent in computing interactions increases as the square of the number of platforms

included in the activity. Considerable improvement in the time performance of simulation may be achieved by heuristics used to reduce the number of computed interactions. One such heuristic would be a structural model of spatial relationsihps that would assign each platform to some region. Computations could then be limited to achieve interactions between platforms in proximate, adjacent, or concurrent regions depending on the platforms and sensors involved.

As might be expected, the organization of WES is intended to be, and succeeds in being, quite "modular." Another direction for developing WES will be to make the model not only more modular, but modularly flexible. In this approach the simulator would manage not a single model but rather a collection of submodels having only well specified and delimited interactions with each other. Sets of submodels could then be selected by the user at simulation time to satisfy his particular needs, and submodel characterization of behaviors and/or activities could be greatly extended allowing the user some control over the tradeoffs that occur in tactical activity.

In addition to the above developments in how the WES simulation is accomplished, three developments are currently underway to extend the domain of what is simulated. First, in modern naval engagements electronic and sonic sensors are the primary source of tactical information--and misinformation. The apparent fidelity of any model for the tactical decision-making environment depends strongly on its characterization of these sensors and must make extensive provisions for representing electronic countermeasures. Currently in WES, each sensor either delivers perfect information or no information at all. This situation is clearly insufficient for instructional purposes. At a minimum WES should include jamming, blip enhancement, range deception, and false reports. Second, the intelligence of WES platforms needs to be enhanced. A tactician does not deal with platforms, but with the organizations of humans who control them. He uses a specialized language, not simply a limited menu of commands, to evoke from these organizations a wide variety of response. Additionally, of course, the intelligence of platforms is a significant element in the construction of an automated opponent. In its most extended form, the problem of platform intelligence is that of designing a natural language understander for the domain of naval tactics. A more realistic goal is to devise a language to allow student/tacticians to express platform missions, in effect, to devise a programming language for platforms. This development is being undertaken. Third, explicit representation of communication is needed in WES. Currently, platforms react instantly and precisely to orders issued by the user. A more reasonable model of the user's communications environment is being devised to account for lag and other degradations in communications.

The objective of this work is to produce a "suitcase exerciser" that could be used aboard ship, at dockside, or wherever to provide practice in the infinite variety of tactical activity that may be faced by a Navy commander. Such an exerciser would execute 5-10 million instructions per second, would provide the functional equivalent of direct memory access to a million bytes of one micro-second memory, and would cost less than $8,000. We expect the state-of-the-art to be able to support such a development in 5-8 years. Those who are familiar with Alan Kay's notions concerning the DYNABOOK (cf. Horn and Winston, 1975) will not be particularly shocked by these expectations. It should be the responsibility of military training developers to be prepared to capitalize on these capabilities when they became available.

Directions for Development

Finally, it may be appropriate to indicate directions that development of intelligent instructional systems might take. There appear to be seven such directions.

1. Representing Knowledge/Skill Areas. The most difficult aspect in the current state-of-the-art is designing adequate representations of subject matter. These representations can be designed relatively easily for knowledge/skill areas that can be generated from a formal grammar. However, subject matter of this sort, although important, is relatively rare. A key objective should be to determine how far computerized representations can be extended into less formal but more common knowledge/skill areas.

2. Entering the Model--Generative Authoring. Adequate, computerized representations of knowledge/skill areas and accompanying instructional strategies must currently be constructed by experienced programmers who are trained in artificial intelligence techniques and who use a powerful list processing language such as LISP or LEAP/SAIL. Using author prompting techniques, it may ultimately be possible for a subject matter specialist with no knowledge of computing to create instructional material by engaging in a computer-initiated dialogue about the subject matter. The computer system would interrogate the subject matter specialist until his knowledge of the subject matter was exhausted or a representation of the subject matter was sufficiently complete to support mixed-initiative, tutorial dialogue and/or tutorial simulation. The computer system would consult as many subject matter specialists or other information sources as desired, and it would be able to provide appropriate, relevant instruction despite major gaps in its knowledge base. Finally, it may be possible for the computer system to construct an appropriate representation of knowledge from standard, already prepared textual material.

3. Natural Language Processing. Because this approach to instruction must support tutorial, student-computer dialogue without the help of pre-specified, human-prepared items and problem statements, computer analysis and synthesis of natural language is an important, relevant problem area. Much of this work is being undertaken outside of the CBI community, but we should remain alert to existing capabilities that can be appropriately applied and evaluated within the context of instruction.

4. Representing Learner Capabilities and Needs. Because this approach to instruction is based on explicit models of the subject matter, it can precisely identify what the learner has and has not mastered. To some extent, this information may serve as a model of the learner. One objective should be to evaluate how adequate this information is for prescribing instructional strategy and in what ways it may need to be modified and/or augmented.

5. Selecting Appropriate Instructional Strategies. This approach to instruction provides a maximally flexible set of treatments and strategies. An objective should be to evaluate ways to use these capabilities to

optimize instructional outcomes subject to such constraints as time available, nature of the input population, nature of the subject matter, and desired distribution of instruction achievement.

6. Generating Instructional Items and Problems. Given appropriate instructional strategy and representation of a knowledge/skill area, the problem of generating specific instructional content for the student remains. General techniques for generating items, problems, and scenarios for simulation as well as techniques for tailoring items, problems, and scenarios to meet the needs of specific students--or groups of students in team, unit, and crew instruction.

7. Computer Facilities. Large, powerful computer systems are currently required for this approach to instruction. However, small computer systems suitable for distributed training may be more appropriate for many uses. An objective should be, therefore, to apply and evaluate capabilities for delivering this type of instruction on smaller computer systems. Current, rapid developments in microcomputer technology make it probable that large, powerful, and complex CBI systems in operational settings may be supported by small stand-alone terminals in 3-5 years.

Final Word

In summary, intelligent instructional systems can be distinguished from more conventional approaches by the automation of instructional interaction and choice of strategy. This approach promises to reduce the costs of instructional materials preparation and to increase the adaptability and individualization of the instruction delivered. The principal instructional capabilities to be evaluated are tutorial simulation and tutorial dialogue. These capabilities require a computer to (1) generate problem statements and solutions, (2) determine efficient sequences, and (3) simulate a variety of situations encountered on the job. The capabilities enable a student to (1) test his own hypotheses concerning the subject matter, (2) probe for information at different levels of difficulty and abstraction, (3) acquire wide experience in minimum time, (4) obtain instructional material generated for his unique abilities and needs, (5) receive instructional aids for partially completed solutions, and (6) receive reviews and critiques of completed problem solutions. These approaches represent real advances in the development of CBI techniques, they are practical, and they are worth pursuing.

REFERENCES

Abelson, R. P., & Reich, C. M. Implicational molecules: A method for extracting meaning from input sentences. In D. E. Walker and L. M. Norton (Eds.). Proceeding of the [First] International Joint Conference on Artificial Intelligence. Bedford, MA: The MITRE Corporation, 1969.

Barr, A., Beard, M., & Atkinson, R. C. A rationale and description of a CAI program to teach the BASIC programming language. Instructional Science, 1975, 4, 1-31.

Brown, J. S., Burton, R. R., & Bell, A. B. SOPHIE: A sophisticated instructional environment for teaching electronic troubleshooting (An example of AI in CAI) (BBN Rep. 2790). Cambridge, MA: Bolt Beranek and Newman, Inc., March 1974.

Brown, J. S., Burton, R. R., & Bell, A. G. SOPHIE: A step toward creating a reactive learning environment. International Journal of Man-Machine Studies, 1975, 7, 675-696.

Brown, J. S., Rubinstein, R., & Burton, R. R. Reactive learning environment for computer assisted electronics instruction (BBN Rep. 3314). Cambridge, MA: Bolt Beranek and Newman, Inc., October 1976.

Carbonell, J. R. AI in CAI: An artificial intelligence approach to computer-assisted instruction. IEEE Transactions on Man-Machine Systems, 1970, 11, 190-202.

Collins, A., Warnock, E. H., & Passafiume, J. J. Analysis and syntheses of tutorial dialogues (Tech. Rep. 2789). Cambridge, MA: Bolt Beranek and Newman, Inc., March 1974

Fletcher, J. D. Computer applications in education and training: Status and trends (NPRDC Tech. Rep. 75-32). San Diego, CA: Navy Personnel Research and Development Center, April 1975.

Ford, J. D., Jr., Slough, D. A., & Hurlock, R. E. Computer-assisted instruction in Navy technical training using a small dedicated computer system: Final report (Res. Rep. SRR 73-13). San Diego, CA: Naval Personnel and Training Research Laboratory, November 1972.

Goldberg, A. Computer-assisted instruction: The application of theorem-proving to adaptive response analysis (Tech. Rep. 203). Stanford, CA: Stanford University, Institute for Mathematical Studies in the Social Sciences, May 1973.[2]

Goldberg, A., & Suppes, P. Computer-assisted instruction in elementary logic at the university level (Tech. Rep. 239). Stanford, CA: Stanford University, Institute for Mathematical Studies in the Social Sciences, November 1974.

Horn, B. K. P., & Winston, P. H. Personal computers. Datamation, 1975, 21, 111-115.

Hurlock, R. E., & Slough, D. A. Experimental evaluation of PLATO IV technology: Final report (NPRDC TR 76TQ-44). San Diego, CA: Navy Personnel Research and Development Center, August 1976.

Kimball, R. B. Self-optimizing computer-assisted tutoring: Theory and practice (Tech. Rep. 206). Stanford, CA: Stanford University, Institute for Mathematical Studies in the Social Sciences, June 1973.

Koffman, E. B., & Blount, S. E. Artificial intelligence and automatic programming in CAI. Artificial Intelligence, 1975, 6, 215-234.

Norman, D. A. Memory, knowledge, and the answering of questions. In R. L. Solso (Ed.) Contemporary Issues in Cognitive Psychology: The Loyola Symposium. Washington, D.C.: V. H. Winston and Sons, 1973.

Ruth, G. R. Analysis of algorithm implementation (MAC TR-130). Cambridge, MA: Massachusetts Institute of Technology, May 1974.

FOOTNOTES

[1] An early version of this paper was presented at the Association for Development of Computer Based Instruction Summer Meeting, August 1975.

[2] These reports are dissertation studies and should be available from University microfilms.

Disclaimer: The opinions and assertions contained herein are those of the writer(s) and are not to be construed as official or reflecting the views of the Navy Department.